Documentation

Hello.c

It is a simple Loadable Kernel Module that shows how we can make our program to run in Kernel space.

In LKM we register our method using module_init() method. When module is loaded this registered method executes. We are just printing “Hello World” Message when module is loaded.

In the same way we have registered our exit method using module_exit(). Our exit method is called when module is unloaded. In this method we are printing “Good Bye”

Sys_call_hooking.c

In this module we are hooking **open()** system call. When our module is loaded, it will simply print message “Your Open() system call is hooked” on open system call.

For example if user enter **vim f1.txt** or **cat f1.txt** etc. our module will print message of hooking.

Creation of Loadable Kernel Module from C (e.g. hello.c) file

1. Place C file (hello.c) and Makefile in same folder
2. Make
3. Hello.ko and some other files will be created

```
Zahid:- ls
hello.c  Makefile
Zahid:- make
make -C /lib/modules/4.10.0-28-generic/build M=/home/zahid/SP/project/pp/system-
call-hooking-linux/hello_world_LKM modules
make[1]: Entering directory '/usr/src/linux-headers-4.10.0-28-generic'
  CC [M]  /home/zahid/SP/project/pp/system-call-hooking-linux/hello_world_LKM/he
llo.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/zahid/SP/project/pp/system-call-hooking-linux/hello_world_LKM/he
llo.mod.o
  LD [M]  /home/zahid/SP/project/pp/system-call-hooking-linux/hello_world_LKM/he
llo.ko
make[1]: Leaving directory '/usr/src/linux-headers-4.10.0-28-generic'
Zahid:- ls
hello.c   hello.mod.c  hello.o   modules.order
hello.ko  hello.mod.o  Makefile  Module.symvers
Zahid:-
```

Figure 1

Loading of Loadable Kernel Module

- sudo insmod hello.ko (Figure 2)

To View our loaded module

- lsmod

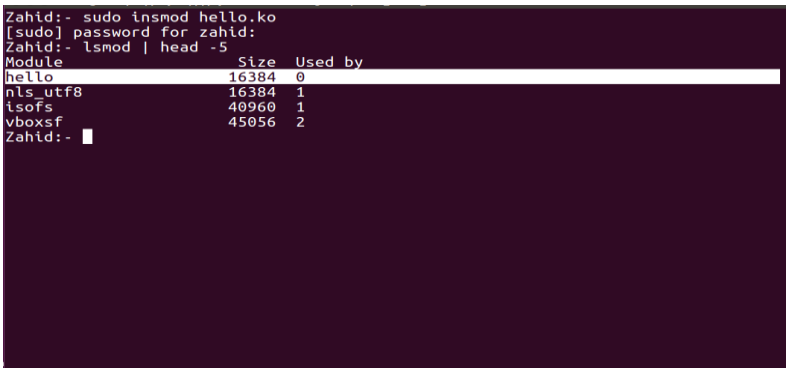

Figure 2

To view message printed by LKM

- dmesg

To clear messages from console

- sudo dmesg -c

To unload LKM

- sudo rmmod hello

Complete demonstration video is available at

https://drive.google.com/drive/u/3/my-drive